

1 38. The system of claim 37, wherein the central processing unit includes the
2 hardware unit.

1 39. The system of claim 37, wherein the hardware unit is outside of the central
2 processing unit.

1 40. The system of claim 37, wherein a portion of the operand stack is stored in
2 the register file of the central processing unit and wherein the hardware unit is adapted to
3 produce at least one of overflow or underflow indications for the portion of the operand
4 stack stored in the register file.

1 41. The system of claim 37, wherein the hardware unit is further adapted to
2 store at least some Java registers in the register file.

1 42. The system of claim 37, wherein the hardware unit implements at least part
2 of a Java virtual machine.

1 43. The system of claim 37, wherein the hardware unit is connected between a
2 memory and the central processing unit.

1 44. The system of claim 37, wherein the hardware unit is adapted to examine the
2 stack-based instructions to determine whether multiple stack-based instructions can be
3 combined into fewer register-based instructions.

1 45. The system of claim 44, wherein the hardware unit produces register-based
2 instructions that access the Java variables in the register file so as to reduce the number of
3 register-based instructions that would otherwise be required.



21839

4 46. The system of claim 44, wherein multiple stack-based instructions pass
5 through the hardware unit concurrently to allow for the operation of the combining logic.

1 47. The system of claim 44, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 48. The system of claim 37, wherein the hardware unit produces an exception upon
2 at least one of the stack-based instructions, and wherein the central processing unit will, in
3 software, translate the at least one of the stack-based instructions causing the exception.

1 49. The system of claim 37, wherein the hardware unit is adapted to swap Java
2 variables in and out of the register file from a memory.

1 50. The system of claim 37, wherein the hardware unit includes logic that keeps
2 track of Java Variables stored in the register file and when a Java bytecode to be translated
3 references a Java Variable stored in a register of the register file, the hardware unit
4 produces an indication of that register to be used in the translation process.

1 51. The system of claim 37, wherein the hardware unit keeps track of which
2 registers in the register file contain Java variables, the meaning of the registers being able
3 to change as a result of an executed instruction.

1 52. A system comprising:
2 a central processing unit having a register file, the central processing unit adapted to
3 execute register-based instructions; and
4 a hardware unit associated with the central processing unit, the hardware unit
5 adapted to convert Java bytecode instructions into register-based instructions, wherein the



21839

6 hardware unit is adapted to store at least one Java register in the central processing unit's
7 register file, wherein at least one of the register-based instructions reference a register in
8 the central processing unit's register file containing one of the at least one Java register, the
9 at least one Java Register including the Java Program Counter.

1 53. The system of claim 52, wherein the central processing unit includes the
2 hardware unit.

1 54. The system of claim 52, wherein the hardware unit is outside of the central
2 processing unit.

1 55. The system of claim 52, wherein the central processing unit includes the
2 hardware unit.

1 56. The system of claim 52, wherein the hardware unit is outside of the central
2 processing unit.

1 57. The system of claim 52, wherein a portion of the operand stack is stored in
2 the register file of the central processing unit and wherein the hardware unit is adapted to
3 produce at least one of overflow or underflow indications for the portion of the operand
4 stack stored in the register file.

1 58. The system of claim 52, wherein the hardware unit is further adapted to
2 store at least some Java variables in the register file.

1 59. The system of claim 52, wherein the hardware unit implements at least part
2 of a Java virtual machine.



21839

1 60. The system of claim 52, wherein the hardware unit is connected between a
2 memory and the central processing unit.

1 61. The system of claim 52, wherein the hardware unit is adapted to examine the
2 stack-based instructions to determine whether multiple stack-based instructions can be
3 combined into fewer register-based instructions.

1 62. The system of claim 61, wherein the hardware unit produces register-based
2 instructions that access the Java registers in the register file so as to reduce the number of
3 register-based instructions that would otherwise be required.

1 63. The system of claim 61, wherein multiple stack-based instructions pass
2 through the hardware unit concurrently to allow for the operation of the combining logic.

1 64. The system of claim 52, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 65. The system of claim 52, wherein the hardware unit produces an exception
2 upon at least one of the stack-based instructions, and wherein the central processing unit
3 will, in software, translate the at least one of the stack-based instructions causing the
4 exception.

1 66. The system of claim 52, wherein the hardware unit is adapted to swap Java
2 registers in and out of the register file from a memory.



21839

1 67. A central processing unit comprising:
2 an input adapted to receive Java bytecode instructions;
3 a register file adapted to be manipulated using register-based instructions; and
4 a hardware subunit adapted to convert Java instructions into register-based
5 instructions, wherein the hardware subunit is adapted to store at least one Java variable in
6 the register file at a location separate from any operand stack, wherein at least one of the
7 register-based instructions reference a register in the central processing unit's register file
8 containing one of the at least one Java variable.

1 68. A central processing unit comprising:
2 an input adapted to receive Java bytecode instructions;
3 a register file adapted to be manipulated using register-based instructions; and
4 a hardware subunit adapted to convert Java instructions into register-based
5 instructions, wherein the hardware subunit is adapted to store at least one Java register in
6 the register file, wherein at least one of the register-based instructions reference a register
7 in the central processing unit's register file containing one of the at least one Java register,
8 the at least one Java Register including the Java Program Counter.

1 69. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit having an
3 associated register file; and
4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 stack-based instructions into register-based instructions, the hardware unit adapted to store
6 portions of the operand stack in the first unit's register file, wherein the hardware unit
7 includes logic to keep a count of how many entries have been placed on the operand stack.



21839

1 70. The system of claim 69, wherein the first unit and the hardware unit are part of
2 a central processing unit.

1 71. The system of claim 69, wherein the first unit comprises a central processing
2 unit and wherein the hardware unit is outside of the central processing unit.

1 72. The system of claim 69, wherein a portion of the operand stack is stored in
2 the register file of the first unit and wherein the hardware unit is adapted to produce at least
3 one of overflow or underflow indications for the portion of the operand stack stored in the
4 register file.

1 73. The system of claim 69, wherein the hardware unit is further adapted to
2 store at least some Java registers in the register file.

1 74. The system of claim 69, wherein the hardware unit implements at least part
2 of a Java virtual machine.

1 75. The system of claim 69, wherein the hardware unit is connected between a
2 memory and the first unit.

1 76. The system of claim 69, wherein the hardware unit is adapted to manage a
2 Java stack.

1 77. The system of claim 69, wherein the hardware unit is adapted to examine the
2 stack-based instructions to determine whether multiple stack-based instructions can be
3 combined into fewer register-based instructions.



21839

1 78. The system of claim 77, wherein the hardware unit produces register-based
2 instructions that access the portion of the operand stack in the register file so as to reduce
3 the number of register-based instructions that would otherwise be required.

1 79. The system of claim 77, wherein multiple stack-based instructions pass
2 through the hardware unit concurrently to allow for the operation of the combining logic.

1 80. The system of claim 69, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 81. The system of claim 80, wherein the multiple Java bytecodes include a basic
2 operand instruction and one or more stack manipulation instructions.

1 82. The system of claim 80, wherein the multiple Java bytecodes includes a load
2 or store instruction.

1 83. The system of claim 69, wherein the hardware unit produces an exception
2 upon at least one of the stack-based instructions, and wherein the central processing unit
3 will, in software, translate the at least one of the stack-based instructions causing the
4 exception.

1 84. The system of claim 69, wherein the hardware unit is adapted to swap
2 portions of the operand stack in and out of the register file from a memory.

1 85. The system of claim 69, wherein the hardware unit includes an indication of
2 the depth of the portion of operand stack stored in the register file.



21839

1 86. The system of claim 69, wherein the hardware unit includes logic to keep a
2 count of how many entries have been placed on the operand stack.

1 87. The system of claim 69, wherein the hardware unit includes logic that keeps
2 track of portions of the Java operand stack stored in the first unit's register file and when a
3 Java bytecode to be translated references an element of the Java operand stack stored in a
4 register of the first unit's register file, the hardware unit produces an indication of that
5 register to be used in the translation process.

1 88. The system of claim 69, wherein the hardware unit keeps track of a top of
2 stack register location, wherein the top of stack register in the first unit's register file is not
3 fixed and can change as a result of an executed instruction.

1 89. The system of claim 69, wherein a portion of the Java operand stack or Java
2 variables are stored in the register file of the first unit, wherein the hardware unit keeps
3 track of which registers in the first unit's register file contains portions of the Java operand
4 stack or Java variables, the meaning of the registers being able to change as a result of an
5 executed instruction.

1 90. The central processing unit of claim 69, wherein an overflow or underflow
2 produces operand transfer between the register file in the central processing unit and
3 memory.

1 91. The central processing unit of claim 69, wherein the registers of the register
2 file of the central processing unit used to store the portion of operand stack is full of valid
3 data.



21839

1 92. The central processing unit of claim 69, wherein the at least one of the
2 overflow or underflow indications is generated by a stack instruction pushing an operand or
3 popping the operand from the operand stack.

1 93. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit having an
3 associated register file; and
4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 Java bytecodes into register-based instructions, wherein the hardware unit includes logic
6 that keeps track of portions of the Java operand stack or Java Variables stored in the first
7 unit's register file and when a Java bytecode to be translated references an element of the
8 Java operand stack or Java Variable stored in a register of the first unit's register file, the
9 hardware unit produces an indication of that register to be used in the translation process.

1 94. The system of claim 93, wherein the first unit and the hardware unit are part
2 of a central processing unit.

1 95. The system of claim 93, wherein the first unit comprises a central processing
2 unit and wherein hardware unit is outside of the central processing unit.

1 96. The system of claim 93, wherein a portion of the operand stack is stored in
2 the register file of the first unit and wherein the hardware unit is adapted to produce at least
3 one of overflow or underflow indications for the portion of the operand stack stored in the
4 register file.

1 97. The system of claim 93, wherein the hardware unit is further adapted to
2 store at least some Java registers in the register file.



21839

1 98. The system of claim 93, wherein the hardware unit implements at least part
2 of a Java virtual machine.

1 99. The system of claim 93, wherein the hardware unit is connected between a
2 memory and the first unit.

1 100. The system of claim 93, wherein the hardware unit is adapted to examine the
2 stack-based instructions to determine whether multiple stack-based instructions can be
3 combined into fewer register-based instructions.

1 101. The system of claim 100, wherein the hardware unit produces register-based
2 instructions that access the portion of the operand stack in the register file so as to reduce
3 the number of register-based instructions that would otherwise be required.

1 102. The system of claim 100, wherein multiple stack-based instructions pass
2 through the hardware unit concurrently to allow for the operation of the combining logic.

1 103. The system of claim 93, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 104. The system of claim 103, wherein the multiple Java bytecodes include a
2 basic operand instruction and one or more stack manipulation instructions.

1 105. The system of claim 103, wherein the multiple Java bytecodes includes a
2 load or store instruction.



21839

1 106. The system of claim 93, wherein the hardware unit produces an exception
2 upon at least one of the stack-based instructions, and wherein the first unit will, in
3 software, translate the at least one of the stack-based instructions causing the exception.

1 107. The system of claim 93, wherein the hardware unit is adapted to swap Java
2 variables or portions of the operand stack in and out of the register file from a memory.

1 108. The system of claim 93, wherein the hardware unit includes an indication of
2 the depth of the portion of operand stack stored in the register file.

1 109. The system of claim 93, wherein the hardware unit includes logic to keep a
2 count of how many entries have been placed on the operand stack.

1 110. The system of claim 93, wherein the hardware unit includes logic that keeps
2 track of Java Variables stored in the first unit's register file and when a Java bytecode to be
3 translated references a Java Variable stored in a register of the first unit's register file, the
4 hardware unit produces an indication of that register to be used in the translation process.

5 111. The system of claim 93, wherein the hardware unit includes logic that keeps
6 track of portions of the Java operand stack stored in the first unit's register file and when a
7 Java bytecode to be translated references an element of the Java operand stack stored in a
8 register of the first unit's register file, the hardware unit produces an indication of that
9 register to be used in the translation process.

1 112. The system of claim 93, wherein the hardware unit keeps track of a top of
2 stack register location, wherein the top of stack register in the first unit's register file is not
3 fixed and can change as a result of an executed instruction.



21839

4 113. The system of claim 93, wherein the hardware unit keeps track of which
5 registers in the first unit's register file contains portions of the Java operand stack or Java
6 variables, the meaning of the registers being able to change as a result of an executed
instruction.

1 114. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit having an
3 associated register file; and
4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 Java bytecodes into register-based instructions, wherein the hardware unit includes logic
6 that keeps track of Java Variables stored in the first unit's register file and when a Java
7 bytecode to be translated references a Java Variable stored in a register of the first unit's
8 register file, the hardware unit produces an indication of that register to be used in the
9 translation process.

1 115. The system of claim 114, wherein the first unit and the hardware unit are
2 part of a central processing unit.

1 116. The system of claim 114, wherein the first unit comprises a central
2 processing unit and wherein the hardware unit is outside of the central processing unit.

1 117. The system of claim 114, wherein the hardware unit is further adapted to
2 store at least some Java registers in the register file.

1 118. The system of claim 114, wherein the hardware unit is adapted to swap Java
2 variables in and out of the register file from a memory.

1 119. A system comprising:



21839

2 a first unit adapted to execute register-based instructions, the first unit having an
3 associated register file; and

4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 Java bytecodes into register-based instructions, wherein the hardware unit includes logic
6 that keeps track of portions of the Java operand stack stored in the first unit's register file
7 and when a Java bytecode to be translated references an element of the Java operand stack
8 stored in a register of the first unit's register file, the hardware unit produces an indication
9 of that register to be used in the translation process.

10 120. The system of claim 119, wherein the first unit and the hardware unit
11 comprise a central processing unit.

12 121. The system of claim 119, wherein the first unit comprises a central
13 processing unit and wherein the hardware unit is outside of the central processing unit.

14 122. The system of claim 119, wherein a portion of the operand stack is stored in
15 the register file of the central processing unit and wherein the hardware unit is adapted to
16 produce at least one of overflow or underflow indications for the portion of the operand
17 stack stored in the register file.

18 123. The system of claim 119, wherein the hardware unit is adapted to examine
19 the stack-based instructions to determine whether multiple stack-based instructions can be
20 combined into fewer register-based instructions.

21 124. The system of claim 119, wherein the hardware unit produces register-based
22 instructions that access the portion of the operand stack in the register file so as to reduce
23 the number of register-based instructions that would otherwise be required.



21839

1 125. The system of claim 119, wherein multiple stack-based instructions pass
2 through the hardware unit concurrently to allow for the operation of the combining logic.

1 126. The system of claim 119, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 127. The system of claim 119, wherein the multiple Java bytecodes include a
2 basic operand instruction and one or more stack manipulation instructions.

1 128. The system of claim 119, wherein the hardware unit includes an indication
2 of the depth of the portion of operand stack stored in the register file.

1 129. The system of claim 119, wherein the hardware unit includes logic to keep a
2 count of how many entries have been placed on the operand stack.

1 130. The system of claim 119, wherein the hardware unit includes logic that keeps
2 track of portions of the Java operand stack stored in the first unit's register file and when a
3 Java bytecode to be translated references an element of the Java operand stack stored in a
4 register of the first unit's register file, the hardware unit produces an indication of that
5 register to be used in the translation process.

1 131. The system of claim 119, wherein the hardware unit keeps track of a top of
2 stack register location, wherein the top of stack register in the first unit's register file is not
3 fixed and can change as a result of an executed instruction.

1 132. The system of claim 119, wherein a portion of the Java operand stack or
2 Java variables are stored in the register file of the first unit, wherein the hardware unit



21839

3 keeps track of which registers in the first unit's register file contains portions of the Java
4 operand stack or Java variables, the meaning of the registers being able to change as a
5 result of an executed instruction.

1 133. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit having an
3 associated register file; and
4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 Java bytecodes into register-based instructions, wherein a portion of the operand stack is
6 stored in the register file of the first unit, wherein the hardware unit keeps track of a top of
7 stack register location, wherein the top of stack register in the first unit's register file is not
8 fixed and can change as a result of an executed instruction.

1 134. The system of claim 133, wherein the first unit and the hardware unit
2 includes the central processing unit.

1 135. The system of claim 133, wherein the first unit comprises a central
2 processing unit and wherein the hardware unit is outside of the central processing unit.

1 136. The system of claim 133, wherein the hardware unit is adapted to produce at
2 least one of overflow or underflow indications for the portion of the operand stack stored in
3 the register file.

1 137. The system of claim 133, wherein the hardware unit is adapted to examine
2 the stack-based instructions to determine whether multiple stack-based instructions can be
3 combined into fewer register-based instructions.



21839

1 138. The system of claim 137, wherein the hardware unit produces register-based
2 instructions that access the portion of the operand stack in the register file so as to reduce
3 the number of register-based instructions that would otherwise be required.

1 139. The system of claim 137, wherein multiple stack-based instructions pass
2 through the hardware unit concurrently to allow for the operation of the combining logic.

1 140. The system of claim 133, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 141. The system of claim 133, wherein the hardware unit produces an exception
2 upon at least one of the stack-based instructions, and wherein the central processing unit
3 will, in software, translate the at least one of the stack-based instructions causing the
4 exception.

1 142. The system of claim 133, wherein the hardware unit includes an indication
2 of the depth of the portion of operand stack stored in the register file.

1 143. The system of claim 133, wherein the hardware unit includes logic to keep a
2 count of how many entries have been placed on the operand stack.

1 144. The system of claim 133, wherein the hardware unit includes logic that keeps
2 track of portions of the Java operand stack stored in the first unit's register file and when a
3 Java bytecode to be translated references an element of the Java operand stack stored in a
4 register of the first unit's register file, the hardware unit produces an indication of that
5 register to be used in the translation process.



21839

1 145. The system of claim 133, wherein the hardware unit keeps track of which
2 registers in the first unit's register file contains portions of the Java operand stack or Java
3 variables, the meaning of the registers being able to change as a result of an executed
4 instruction.

1 146. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit having an
3 associated register file; and
4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 Java bytecodes into register-based instructions, wherein a portion of the Java operand stack
6 or Java variables are stored in the register file of the first unit, wherein the hardware unit
7 keeps track of which registers in the first unit's register file contains portions of the Java
8 operand stack or Java variables, the meaning of the registers being able to change as a
9 result of an executed instruction.

1 147. The system of claim 146, wherein a top of stack register in the first unit's
2 register file is not fixed and can change as a result of an executed instruction.

1 148. The system of claim 146, wherein the first unit and the hardware unit
2 comprise a central processing unit.

1 149. The system of claim 146, wherein the first unit comprises a central
2 processing unit and the hardware unit is outside of the central processing unit.

1 150. The system of claim 146, wherein the hardware unit is adapted to produce at
2 least one of overflow or underflow indications for the portion of the operand stack stored in
3 the register file.



21839

1 151. The system of claim 146, wherein the hardware unit is adapted to examine
2 the stack-based instructions to determine whether multiple stack-based instructions can be
3 combined into fewer register-based instructions.

1 152. The system of claim 151, wherein the hardware unit produces register-based
2 instructions that access the portion of the operand stack in the register file so as to reduce
3 the number of register-based instructions that would otherwise be required.

1 153. The system of claim 151, wherein multiple stack-based instructions pass
2 through the hardware unit concurrently to allow for the operation of the combining logic.

1 154. The system of claim 146, wherein the hardware unit is adapted to convert
2 multiple Java bytecodes into a single register-based instruction.

1 155. The system of claim 154, wherein the multiple Java bytecodes include a
2 basic operand instruction and one or more stack manipulation instructions.

1 156. The system of claim 146, wherein the hardware unit includes an indication
2 of the depth of the portion of operand stack stored in the register file.

1 157. The system of claim 146, wherein the hardware unit includes logic to keep a
2 count of how many entries have been placed on the operand stack.

1 158. The system of claim 146, wherein the hardware unit includes logic that keeps
2 track of portions of the Java operand stack stored in the first unit's register file and when a
3 Java bytecode to be translated references an element of the Java operand stack stored in a



21839

4 register of the first unit's register file, the hardware unit produces an indication of that
5 register to be used in the translation process.

1 159. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit having at
3 least two associated register files of registers, the first set of register files used for normal
4 operation, the second set of register files used for operation in a hardware translation mode;
5 and
6 a hardware unit associated with the first unit, the hardware unit adapted to convert
7 stack-based instructions into register-based instructions during the hardware translation
8 mode.

1 160. A system comprising:
2 a first unit adapted to execute register-based instructions, the first unit adapted to
3 produce a "branch taken" indication; and
4 a hardware unit associated with the first unit, the hardware unit adapted to convert
5 stack-based instructions into register-based instructions, the hardware unit containing
6 multiple pipeline stages, the pipeline stages being flushed when the "branch taken"
7 indication is produced.

1 161. A system comprising:
2 a first unit adapted to execute register-based instructions; and
3 a hardware unit associated with the first unit, the hardware unit adapted to convert
4 stack-based instructions into register-based instructions, the hardware unit including a
5 stack-based instruction buffer associated with a decode unit.



21839

1 162. The system of claim 161, wherein decode unit can select multiple stack-
2 based instructions to decode at one time.

1 163. A system comprising:
2 a first unit adapted to execute register-based instructions; and
3 a hardware unit associated with the first unit, the hardware unit adapted to convert
4 stack-based instructions into register-based instructions, the hardware unit including a
5 stack-based instruction buffer associated with a decode unit, the hardware unit adapted to
6 rearrange the stack-based instructions in the buffer to make them easier to be operated on
7 by a decode unit.

1 164. The system of claim 164 wherein decode unit can select multiple stack-based
2 instructions to decode at one time.

1 165. The system of claim 164 wherein the any stack-based instruction or
2 instructions sent to the decoder unit for translation are removed from the stack-based
3 instruction buffer and new stack-based instruction or instructions added to the buffer.

1 166. A system comprising:
2 a first unit adapted to execute register-based instructions; and
3 a hardware unit associated with the first unit, the hardware unit adapted to convert
4 stack-based instructions into register-based instructions, wherein certain instructions are not
5 translated in the hardware unit but instead cause translation software to be loaded into the
6 first unit.



21839

09938865 032481-034

1 167. A system comprising:
2 a central processing unit having a register file, the central processing unit adapted to
3 execute register-based instructions; and
4 a hardware unit associated with the central processing unit, the hardware unit
5 adapted to convert stack-based instructions into register-based instructions, wherein a
6 portion of the operand stack is stored in the register file of the central processing unit and
7 wherein the hardware unit is adapted to produce at least one of overflow or underflow
8 indications for the portion of the operand stack stored in the register file.

1 168. The system of claim 167, wherein the central processing unit includes the
2 hardware unit.

1 169. The system of claim 167, wherein the hardware unit is outside of the central
2 processing unit.

1 170. The system of claim 167, wherein the hardware unit is adapted to swap parts
2 of the operand stack in and out of the register file from a memory.

1 171. The system of claim 167, wherein the system includes an indication of the
2 depth of the portion of operand stack stored in the register file.

1 172. The system of claim 167, wherein the indication of the operand stack depth
2 is stored in the hardware unit.

1 173. The system of claim 167, wherein a overflow or underflow produces
2 operand transfer between the register file in the central processing unit and memory.



21839

1 174. The system of claim 167, wherein, the stack based instructions are Java
2 bytecodes

1 175. The system of claim 167, wherein the registers of the register file of the
2 central processing unit used to store the portion of operand stack is full of valid data.

1 176. The system of claim 175, wherein the at least one of the overflow or
2 underflow indications is generated by a stack instruction pushing an operand or popping the
3 operand from the operand stack.

1 177. The system of Claim 167, wherein the hardware unit has access to the data bus
2 of the central processing unit.

1 178. The system of Claim 167, wherein the hardware unit is further adapted to store
2 at least some Java variables in the register file.

1 179. The system of Claim 167, wherein the hardware unit is further adapted to store
2 at least some Java registers in the register file.

1 180. The system of Claim 167, wherein the stack-based instructions are associated
2 with a virtual machine.

1 181. The system of Claim 167, wherein the stack-based instructions are Java
2 bytecode.

1 182. The system of Claim 167, wherein the hardware unit implements at least part
2 of a Java virtual machine.



21839

1
21
21
21
2

- 1
- 2
- 3

- 1
- 2
- 3

1
21
2

1 191. The system of Claim 190, wherein the multiple Java bytecodes include a basic
2 operand instruction and one or more stack manipulation instructions.

1 192. The system of Claim 190, wherein the multiple Java bytecodes includes a load
2 or store instruction.

1 193. The system of Claim 167, wherein the central processing unit and hardware unit
2 are on the same chip.

1 194. The system of Claim 167, wherein the hardware unit produces an exception
2 upon at least one of the stack-based instructions, and wherein the central processing unit will,
3 in software, translate the at least one of the stack-based instructions causing the exception.

1 195. The system of claim 167, wherein the hardware unit includes logic to keep a
2 count of how many entries have been placed on the operand stack.

1 196. The system of claim 167, wherein the hardware unit includes logic that
2 keeps track of portions of the Java operand stack stored in the register file and when a Java
3 bytecode to be translated references an element of the Java operand stack stored in a register
4 of the register file, the hardware unit produces an indication of that register to be used in the
5 translation process.

1 197. The system of claim 167, wherein the hardware unit keeps track of a top of
2 stack register location, wherein the top of stack register in the register file is not fixed and can
3 change as a result of an executed instruction.



21839

1 198. The system of claim 167, wherein the hardware subunit keeps track of which
2 registers in the register file contain portions of the Java operand stack, the meaning of the
3 registers being able to change as a result of an executed instruction.

1 199. The system of claim 168, wherein the central processing unit includes an
2 execution unit to execute the register-based instructions.

1 200. The system of claim 167, wherein the translated register-based instructions are
2 produced internally within the central processing unit.

1 201. The system of claim 167, wherein register-based instructions cause the
2 manipulation of the register file.

1 202. A central processing unit comprising
2 an input adapted to receive stack-based instructions;
3 a register file adapted to be manipulated using register-based instructions, the register
4 file adapted to store a portion of an operand stack; and
5 a hardware subunit adapted to convert stack-based instructions into register-based
6 instructions, wherein the hardware subunit is adapted to produce at least one of overflow or
7 underflow indications for the portion of the operand stack stored in the register file.

1 203. The central processing unit of claim 202, wherein the stack-based instructions
2 are Java bytecodes.

1 204. The central processing unit of Claim 202, wherein the hardware subunit is
2 adapted to swap parts of the operand stack in and out of the register file from a memory.



21839

1 205. The central processing unit of Claim 202, wherein the hardware subunit is
2 further adapted to store at least some Java variables in the register file.

1 206. The central processing unit of Claim 202, wherein the hardware subunit is
2 further adapted to store at least some Java registers in the register file.

1 207. The central processing unit of Claim 202, wherein the central processing unit
2 includes an indication of the depth of the portion of operand stack stored in the register file.

1 208. The central processing unit of Claim 202 wherein the indication of the operand
2 stack depth is stored in the hardware subunit.

1 209. The central processing unit of Claim 202 wherein a overflow or underflow
2 produces operand transfer between the register file in the central processing unit and memory.

1 210. The central processing unit of Claim 202 wherein, the stack-based instructions
2 are Java bytecodes.

1 211. The central processing unit of Claim 202, wherein the registers of the register
2 file of the central processing unit used to store the portion of operand stack is full of valid data.

1 212. The central processing unit of Claim 202, wherein the at least one of the
2 overflow or underflow indications is generated by a stack instruction pushing an operand or
3 popping the operand from the operand stack.

1 213. The central processing unit of Claim 202, wherein the stack-based instructions
2 are associated with a virtual machine.



21839

1 214. The central processing unit of Claim 202, wherein the stack-based instructions
2 are Java bytecodes.

1 215. The central processing unit of Claim 202, wherein the hardware subunit
2 implements at least part of a Java virtual machine.

1 216. The central processing unit of Claim 202, wherein the hardware subunit is
2 adapted to manage a Java stack.

1 217. The central processing unit of Claim 202, wherein the hardware subunit has
2 access to at least one bus of the central processing unit.

1 218. The central processing unit of Claim 202, wherein the hardware subunit is
2 adapted to examine the stack-based instructions to determine whether multiple stack-based
3 instructions can be combined into fewer register-based instructions.

1 219. The central processing unit of Claim 218, wherein the hardware subunit
2 produces register-based instructions that access the portion of the operand stack in the register
3 file so as to reduce the number of register-based instructions that would otherwise be required.

1 220. The central processing unit of Claim 218, wherein multiple stack-based
2 instructions pass through the hardware subunit concurrently to allow for the operation of the
3 combining logic.

1 221. The central processing unit of Claim 202, wherein the hardware subunit is
2 adapted to convert multiple Java bytecodes into a single register-based instruction.



21839

1 222. The central processing unit of Claim 221, wherein the multiple Java bytecodes
2 include a basic operand instruction and one or more stack manipulation instructions.

1 223. The central processing unit of Claim 221, wherein the multiple Java bytecodes
2 includes a load or store instruction.

1 224. The central processing unit of Claim 202, wherein the hardware subunit
2 produces an exception upon at least one of the stack-based instructions, and wherein the central
3 processing unit will, in software, translate the at least one of the stack-based instructions
4 causing the exception.

5 225. The system of claim 202, wherein the hardware subunit includes logic to keep
6 a count of how many entries have been placed on the operand stack.

1 226. The system of claim 202, wherein the hardware subunit includes logic that keeps
2 track of Java variables stored in the register file and when a Java bytecode to be translated
3 references a Java variable stored in a register of the register file, the hardware subunit
4 produces an indication of that register to be used in the translation process.

1 227. The system of claim 202, wherein the hardware subunit includes logic that keeps
2 track of portions of the Java operand stack stored in the register file and when a Java bytecode
3 to be translated references an element of the Java operand stack stored in a register of the
4 register file, the hardware unit produces an indication of that register to be used in the
5 translation process.



21839

1 228. The system of claim 202, wherein the hardware subunit keeps track of which
2 registers in the register file contain portions of the Java operand stack, the meaning of the
3 registers being able to change as a result of an executed instruction.

1 229. The system of claim 202, wherein the central processing unit includes an
2 execution unit to execute the register-based instructions.

1 230. The system of claim 202, wherein the translated register-based instructions are
2 produced internally within the central processing unit.

1 231. A system comprising:
2 an execution unit associated with a register file, the execution unit adapted to execute
3 decoded instructions; and

4 hardware adapted to receive Java bytecodes and native non-Java instructions and
5 adapted to produce decoded instructions to the execution unit, the hardware including a java
6 hardware unit adapted to store at least one Java variable in the register file at a location
7 separate from any operand stack, wherein at least one of the decoded instructions reference a
8 register in the central processing unit's register file containing one of the at least one Java
9 Variable, wherein a portion of the operand stack is stored in the register file and wherein the
10 hardware unit is adapted to produce at least one of overflow or underflow indications for the
11 portion of the operand stack stored in the register file.

1 232. The system of claim 231, wherein the execution unit and associated register file
2 are part of a central processing unit.

1 233. The system of claim 232, wherein the java hardware unit is part of the central
2 processing unit.



21839

032481-034

1 234. The system of claim 232, wherein the java hardware unit is outside of the
2 central processing unit.

1 235. The system of claim 234, wherein the java hardware unit is further adapted to
2 translate Java bytecodes into native instructions.

236. The system of claim 235, wherein the hardware includes portions of the central
processing unit, the portions including a decoder.

237. The system of claim 231, wherein the java hardware unit is further adapted to
translate Java bytecodes into native instructions.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: _____

Joseph P. O'Malley
Registration No. 36,226
Redwood Shores, CA office
Tel: (650) 622-2333

P.O. Box 1404
Alexandria, Virginia 22313-1404

Date: August 23, 2001



21839